言語サービス記述のための上位オントロジーの提案

Proposal of an Upper Ontology for Describing Linguistic Services

林 良彦（大阪大学大学院言語文化研究科，独立行政法人 情報通信機構）
Yoshihiko Hayashi, Osaka University and NICT
楢和 千春（京都大学大学院情報学研究科）
Chiharu Narawa, Kyoto University

1 はじめに

近年，さまざまな言語資源や言語処理機能が Web 上で利用可能となっている．テキスト翻訳や辞書アクセスなどの言語にかかわるサービス機能を直接ユーザに提供するサイトのほか，意味概念辞書やオントロジーといった言語資源や，形態素解析などの各種の自然言語処理ツールも Web 上で盛んに公開されている．また一方では，ドメインや目的に特化した言語資源が NPO などのコミュニティによって独自に開発されてきている．もし，これらの要素を異文化コラボレーション等におけるさまざまな利用の局面に応じて効率よく連携させることができれば，ユーザごとの目的に応じた有用な言語的機能を提供できる可能性がある．

言語グリッド[1]は，そのような言語にかかわるサービス機能（以下，言語サービス）を Web 上で提供するための言語基盤である．本報告は，言語グリッドのような言語基盤において言語サービスを記述するためのオントロジーの最上位階層の構成案を提示する．言語サービスには，言語表現の解析や変換などのいわゆる自然言語処理機能だけでなく，辞書などの言語資源のアクセス機能が含まれる．よって，このオントロジーにおいては，言語資源や抽象的な言語オブジェクト，また，それらの間の関係を規定できることが必要となる．

2 言語基盤と言語サービスオントロジー

2.1 言語基盤の目的

言語基盤としての言語グリッドの目的は，次の 2 点に集約される．

- 既存の言語資源や言語処理機能を連携させることにより，利用者のニーズに即した複合的な言語サービスを効率よく実現できるようにする
- コミュニティなどで新たに開発された言語資源やそれに基づく言語処理機能を言語サービスとして公開することを容易にする

[1] http://langrid.nict.go.jp/

前者において，複合的な言語サービスのオーサリングは，現状ではユーザがワークフローを記述することにより行う（Murakami，2006）が，将来的にはプランニングによる自動構築が検討されるべきである．いずれにおいても，複合的な言語サービスを構成する原子的要素に関する適切なプロファイル記述がなければ，適切な連携ワークフローを構成することはできない．一方，このようなプロファイル記述は，後者においても必要である．すなわち，新たに公開する言語サービスが利用されるためには，それが発見され，ワークフローに組み込まれる必要がある．さらに，どちらの場合においても，各要素の詳細を隠蔽し対象の言語基盤において定められた標準的な API を実装するラッパープログラムによりラップされ，言語基盤上に配備されなければならない．

2.2 言語基盤のアーキテクチャ

図 1 に言語グリッドのアーキテクチャの概念図を示す．

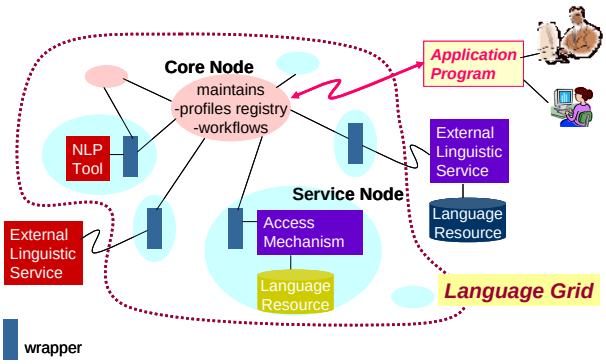

図 1: 言語グリッドのアーキテクチャ

言語グリッドを利用しようとする応用プログラムは，コアノードと呼ばれるノードを介して言語サービスを利用する．コアノードには言語基盤上で公開されている言語サービスに関するプロファイルや複合的なサービスのワークフローが蓄積されており，ワークフローに従って複合的または原子的な言語サービスを実行する．一方，原子的な言語サービスを実行するノードは，サービスノードと呼ば

れる．サービスノードが提供する言語サービスは，そのノード内に実行主体・言語資源を持つ場合もあれば，外部のサービスを仲介するだけの場合もありえる．いずれの場合も，言語基盤で定められた標準的な API を実装するラッパープログラムを介して利用されることになる．

2.3 言語基盤におけるサービスオントロジーの要件

上記のように，言語サービスの必要な構成要素に関しては，適切なプロファイル記述が与えられる必要がある．このプロファイル記述は，少なくとも当該の言語基盤上においては統一された語彙・形式で記述される必要があり，さらには，同等な言語基盤とも相互運用性を持つことが望まれる．このため，共通に理解可能な標準にのっとっていることが必要になる．言語サービスオントロジーは，このような標準化された記述を与えるためのボキャブラリを提供する知識体系であり，それが満たすべき要件は以下のように要約することができる．

- **入出力の制約の表現：** 複合言語サービスを可能にするためには，最初のサービスの出力が次のサービスの入力として引き継がれるように，入出力の互換性をチェックしなければならない．従って，言語サービスオントロジーが満たすべき最も重要な要件は，入出力の制約を十分に記述し得る語彙をもつことである．さらに，将来の自動プランニングによるサービス連携に備えて，言語サービスが呼び出されるときの前提条件や，サービスが呼び出されたときに，もたらされる状態変化の効果を適切に定義しておかなければならない．
- **構成要素を記述するに十分な語彙：** 言語サービスの構成要素，即ち，言語処理機能と言語資源の両方を記述する語彙をもつ必要がある．さらには，言語処理機能によって付加されるさまざまなレベルのアノテーションの記述モデルを提供する必要がある．
- **標準との適合：** 構成要素に関する記述の再利用性・相互運用性を確保するために，記述の体系は国際標準として通用している規格に適合していなければならない．

3 上位オントロジーの構造

3.1 最上位階層

言語サービスオントロジーの最上位階層を図 2 に示す[2]．長方形の枠はクラスを示し，矢印とその横のラベルはクラス間の関係を示す．‘isa’のラベルは，クラス階層を表わしている．

まず，最上位のクラスは，言語に関する広義の資源を包括する **NL_Resource** である．このクラスは，静的な(データ的な)言語資源のための **LangugaeResource** と，プログラム，ツール，システムといった処理プロセス的な資源である **ProcessingResource** へとサブクラス化される．このサブクラス化の考え方は，GATE (Cunningham, 2002) などにおける考え方と同様である．

また，言語サービス **LinguisticService** は，上記の処理資源 ProcessingResource により実現されると規定されている．これは，(静的な)言語資源 LanguageResource は，言語処理資源に属するなんらかのアクセス機能を介して初めて利用可能となることを示している．この場合に当該の ProcessingResource は，LanguageResource を利用するという関係が **uses/usedBy** という属性により表現されている．

また，図 2 の右部では，抽象的な言語オブジェクトとその間の関係が示されている．すなわち，言語表現(**Expression**)は何らかの意味を指示（**denotes**)し，その意味はさらに，意味記述 **Description** により説明される（**describedBy**)．なお，意味がどのような表現により定義されるかについては現在のオントロジーでは規定しない．これにより，プレースホルダとして，ある意味が存在することを表現することを可能とする[3]．

さらに，言語表現には，どのような言語処理が施されているかを表す状態表示（**NLProcessedStatus**）を持つことができ，また，その処理結果は，言語的注釈 **LinguisticAnnotation** により注釈付けられる（**annotatedBy**)．

3.2 言語資源

紙面の関係で，言語資源 LanguageResource の分類体系（タクソノミー）(Hayashi, 2007) を提示することができないが，言語資源はそのタイプによりサブクラス化される．現在のところは，辞書（**Dictionary**）とコーパス（**Corpus**）にサブクラス化される．それぞれのクラスは，記述するデータのタイプ(どのような情報を保持しているか; どのような言語を扱っているか）によって分類される．たとえば，辞書クラスは現在のところ，単言語辞書 **MonolingualDictionary**，対訳辞書 **BilingualDictionary**，多言語用語集 **MultilingualTerminology**，概念辞書 **ConceptLexicon** にサブクラス化されている．

なお，辞書における情報間の関係は，図 3 に示すようにモデル化される．まず，辞書見出しの言語表現（**DictionaryEntryExpression**; Expression の下位クラス)は，辞書的意味 （**DictionaryMeaning**; Meaning の下位クラス)を指示し，さらにその辞書的意

[2] OWL を作業用の記述言語とし，ツールとして Protégé (http:// protege.stanford.edu/) を利用している．

[3] これにより，翻訳やある種の言い換え処理における入出力の言語表現の意味的等価性を示すことを可能とする．

味は，辞書意味記述（**DictionaryDescription**; Description の下位クラス）により説明される．ここで問題となるのが，辞書エントリに記載されているさまざまな情報をどのように分類するかである．より具体的には，あるタイプの情報を意味の表象とみるか，意味記述であるとみるかが問題となる．現在のモデル化においては，いずれのタイプの辞書においても，辞書的意味(DictionaryMeaning)は同義関係にある語の集合によって定義されると考え，その他の豊富な情報(語釈文，用例，語源説明など)は，辞書意味記述に分類されるとしている．この考え方は基本的には，Princeton WordNet における語彙概念に基づく情報モデルによって辞書のモデル化が可能であろう（Hayashi, 2006）という考えに基づいている．

3.3 処理資源

処理資源 ProcessingResource は，入力(**hasInput**)，または，出力(**hasOutput**)の少なくともどちらか一方の値域として言語表現クラス(Expression)をとる．そのタクソノミーの最上位は，抽象的入力器（**AbstractReader**），抽象的出力器(**AbstractWriter**)，言語資源アクセス器（**LR_Accessor**)，言語処理器（**LinguisticProcessor**)の 4 つのサブクラスにより構成される．最初の二つは，音声認識・合成のような非言語メディアとの入出力のために設けられたものである．**LR_Accessor** は，言語資源のアクセスのための処理資源であり，言語表現クラスの下位クラスであるアクセスクエリ **LR_Access_Query** を入力の値域とし，辞書意味クラスを出力の値域とする．このクラスは，さらにアクセスする言語資源のタクソノミーに応じて詳細化される．

言語処理器クラスは現在のところ，言語変換器(**Transformer**)，言語解析器(**Analyzer**)の 2 つのサブクラスに分かれる．前者は主として，言い換え(**Paraphrasor**)や翻訳(**Translator**)の言語変換サービスを扱うためのものである．両クラスとも入力，出力に言語表現クラスをとるが，自明なように入出力の言語の同一性が異なる．現在のオントロジー記述言語として利用している OWL においては，この制約を直接記述することができないので，言い換えにおいては対象言語，翻訳においては対象言語ペアごとにサブクラス化を行う．

図 4 に言語解析器クラスのタクソノミーを示す．現在のところ，まず，チャンク分け(**Chunker**)，品詞付与(**PosTagger**)などの基本機能によってサブクラス化を行い，さらにこれらを基にして，通常用いられているソフトウェアのカテゴリを定義している．たとえば，依存構造に基づく構文解析器(**DependencyParser**)は構文解析器（**Parser**）のサブクラスである．一方，形態素解析器(**MorphologicalAnalyzer**)は，図に示すような 4 つの基本的な解析機能を多重継承するものとして定義している．現在のところ，たとえば厳密な形態素解析器の定義が存在しないこと，また，基本的な解析機能の中身については言語に依存する部分も多いことから，上記に述べた形態素解析器の定義の仕方には議論の余地がある．しかしながら，個々のプロダクトや対象言語の個別性を追求していくと個々のソフトウェアプロダクトのみを記述していくこととなり，形態素解析という一般的であるべき概念を規定することができなくなるという問題がある．

なお，紙面の関係で詳細を紹介できないが，言語処理状態表示(**NLProcessedStatus**)は，部分的に言語処理器と同型なタクソノミーを持つものとして定義されており，各言語処理器の入力に対する前提条件(**hasPrecondition**)や出力が持つ特性(**hasEffect**)を定義するのに用いられる．たとえば，ある構文解析器が形態素解析済みの入力を受け付けるとする．この場合，その構文解析器の入力は，もとより言語表現クラスに制約されているが，さらに，その **hasNLProcessedStatus** 属性の値域は NLProcessedStatus の下位クラスである **MorphologicallyAnalyzed** に制約されるというように定義する．

また，言語処理器，とくに言語解析器は言語表現に対してアノテーションを付与するものであるという近年の考え方に従い，言語解析器の出力である言語表現の annotatedBy 属性の値域として，言語的注釈クラス **LinguisticAnnotation** を指定する．

4 関連研究

自然言語処理サービスのオントロジーを検討した先駆例としては(Klein, 2004)があるが，言語資源のアクセス機能や抽象的な言語オブジェクトの扱いが不十分である．

本検討と密接に関連する標準化関係の動向として，辞書モデリング LMF（Francopoulo, 2006），言語データカテゴリ DCR（Romary, 2004），言語的注釈 LMF（Ide, 2004)がある．これらにおける検討結果を適切に取り入れたり，論理的な整合性をとっていくことが重要である．なお，本検討と多くの動機を共有するプロジェクトとして Lirics[4]プロジェクトがある．

5 おわりに

本報告では，言語グリッドのような言語基盤における言語サービスオントロジーの必要性・要件を示し，オントロジー最上位階層の構成案を提示した．今後は，実際に言語グリッド上で利用が検討されている言語資源，及び，言語処理ツールや言語処理システムの記述を行なう作業を通して，オントロジーの詳細化を図る．また，関連する標準

[4] http://lirics.loria.fr/index.html

化イニシアティブとの連携を模索していく予定である.

参考文献

1. Cunningham, H. et al.: GATE: A Framework and Graphical Development Environment for Robust NLP tools and applications. Proc. of ACL2002, pp.168-175. (2002).
2. Francopoulo, G., et al.: Lexical Markup Framework (LMF). Proc. of LREC2006, pp.233-236. (2006).
3. Hayashi, Y., Ishida, T.: A Dictionary Model for Unifying Machine Readable Dictionaries and Computational Concept Lexicons. Proc. of LREC2006, pp.1-6. (2006).
4. Hayashi,Y.: Conceptual Framework of Upper Ontology for Describing Linguistic Services. Proc. of IWIC2007, pp.246-260.(2007).
5. Ide, N., Romary, L.: International standard for a linguistic annotation framework. Journal of Natural Language Engineering, Vol.10:3-4, pp.211-225. (2004).
6. Klein, E., Potter, S.: An ontology for NLP services. Proc. of LREC Workshop on a Registry of Linguistic Data Categories within an Integrated Language Resource Repository Area. (2004).
7. Murakami, Y.,et al..: Infrastructure for Language Service Composition. Proc. of SKG2006. (2006).
8. Romary, L (eds.): Towards a Data Category Registry for ISO TC37. http://www.tc37sc4.org/new_doc/ISO_TC_37-4_N133_DCR_for_TC37.pdf (2004).

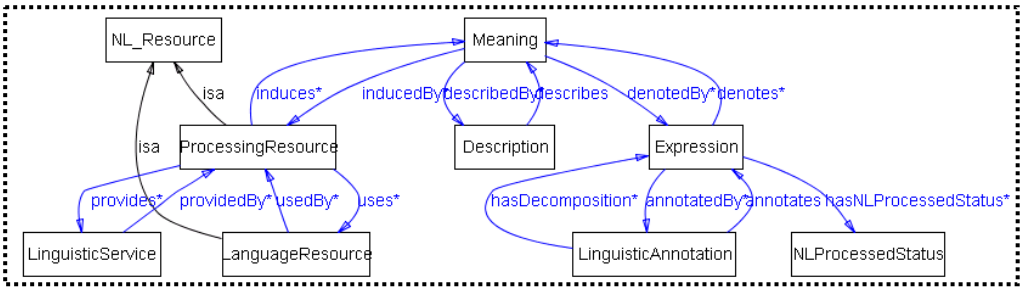

図 2: オントロジーの最上位階層

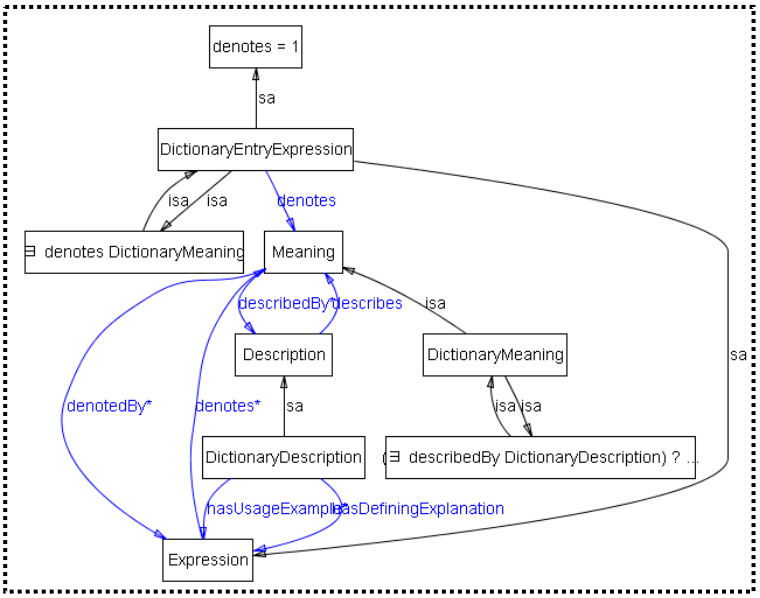

図 3: 辞書における表現・意味・意味記述の関係

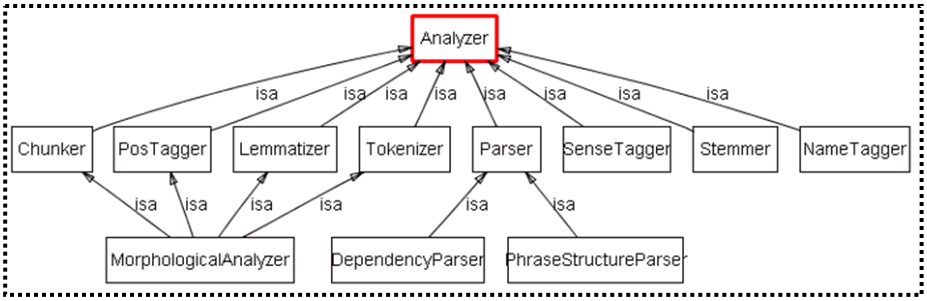

図 4: 言語解析器のタクソノミー試案